Leveraging Biochemical Knowledge Extraction from Academic Literature through Large Language Models: A study of fine-tuning and similarity search

Paulo do Carmo [a,*], Marcos Gôlo [b], Jonas Gwozdz [a], Edgard Marx [a], Stefan Schmidt-Dichte [a], Caterina Thimm [a], Matthias Jooß [a], Pit Fröhlich [a] and Ricardo Marcacini [b]

[a] *Faculty of Informatics, Leipzig University of Applied Sciences (HTWK),Germany*
E-mails: paulo.carmo@htwk-leipzig.de, jonas.gwozdz@htwk-leipzig.de, marx@infai.org
[b] *Institute of Mathematical and Computer Sciences, University of São Paulo (USP),Brazil*
E-mails: marcosgolo@usp.br, ricardo.marcacini@icmc.usp.br

Abstract. The discovery of new drugs based on natural products is related to the efficient extraction of biochemical knowledge from scientific literature. Recent studies have introduced several enhancements to the NatUKE benchmark, improving the performance of knowledge graph embedding methods. These enhancements include refined PDF text extraction, named entity recognition, and improved embedding techniques. Notably, some approaches have incorporated large language models (LLMs). Building on these advances, this study explores the fine-tuning of LLMs with similarity search, exploring both open-source and proprietary models for the automatic extraction of biochemical properties. We fine-tune the LLMs with similarity search to mitigate textual inconsistencies and enhance the prediction of five target properties: compound name, bioactivity, species, collection site, and isolation type. Experimental results demonstrate that similarity search consistently improves the performance, and open-source models can be competitive, occasionally outperforming proprietary models. We also find that the effectiveness of fine-tuning varies across models and biochemical properties. Overall, our findings highlight the potential of LLMs, particularly when fine-tuned and augmented with similarity search, as powerful tools for accelerating the extraction of biochemical knowledge from scientific texts.

Keywords: Large Language Models with similarity search, Finetuning with similarity search, NatUKE Benchmark

1. Introduction

The search for new drugs remains one of the most important and challenging goals in biomedical research [1]. In this context, natural products have played a central role in drug development due to their structural diversity and biological relevance, making these compounds promising candidates in drug discovery pipelines [2]. Between 1981 and 2019, approximately 23.4% of all approved drugs were derived from or inspired by natural products [3], highlighting their ongoing importance in modern pharmacology. However, the identification and characterization of these compounds continue to be a complex and time-consuming process, performed manually by experts [4].

*Corresponding author. E-mail: paulo.carmo@htwk-leipzig.de.

Consequently, there is an increasing need for computational approaches that can efficiently explore the vast body of academic literature to extract relevant biochemical knowledge.

With the growing volume of scientific literature in the field of chemistry containing valuable information about natural products, new opportunities have emerged for integrating and exploring biochemical knowledge [5]. A notable example is $NUBBE_{db}$, a database that compiles Brazilian natural product compounds along with associated biochemical and biological information extracted from scientific articles [5]. This abundance of heterogeneous data has motivated the use of knowledge graphs as a promising strategy for integrating and representing semantic relationships among biochemical entities [6]. In particular, RDF (Resource Description Framework) representation, structuring nodes and edges in the $subject \rightarrow predicate \rightarrow object$ format, enables knowledge modeling in a flexible, interoperable, and extensible manner, as RDF allows knowledge graphs to be composed of data from a single domain [7].

To make the $NUBBE_{db}$ data semantically integrable, do Carmo et al. [6] developed the $NUBBE_{kg}$, which is a knowledge graph that represents, in RDF format, the entities and relationships present in the database. This graph serves as a foundation for knowledge mining and machine learning tasks in the domain of natural products. Based on $NUBBE_{kg}$, the NatUKE benchmark was constructed to evaluate automatic knowledge extraction methods from scientific literature in chemistry. NatUKE consists of a standardized train-test split (gold standard), derived from $NUBBE_{kg}$, aimed at predicting five biochemical properties extracted from scientific articles: *compound name*, *bioactivity*, *species*, *collection site*, and *isolation type*. To enable a robust comparative evaluation, the benchmark has prior execution of four baseline knowledge graph embedding methods [6].

Several previous studies have explored knowledge graph embedding techniques on NatUKE, employing methods such as DeepWalk, node2vec, metapath2vec, and EPHEN [6]. More recent strategies have employed BFS-based searches to enrich the local context before generating embeddings [8]. Other approaches combine named entity recognition with graph embeddings to improve the semantic anchoring of texts to knowledge graphs [9]. Techniques integrating language models such as GPT-3.5 have also been applied during the preprocessing stage of scientific texts for subsequent use with embedding-based methods [10]. Recent advancements, such as the evolution of the EPHEN method [11], demonstrate that the quality of text extraction and the use of embeddings derived from Large Language Models (LLMs) can directly impact performance on the NatUKE benchmark. However, despite the increasing use of LLMs in these pipelines, there remains a lack of approaches that directly employ these models to extract specific biochemical properties from scientific articles.

We propose utilizing LLMs for biochemical property extraction, exploring both open-source and proprietary models in various scenarios, including zero-shot and fine-tuning. While fine-tuning contributes to the better adaptation of models to the biochemical domain, the textual nature of the prediction remains subject to subtle writing errors, such as orthographic variations or the generation of biochemically similar but incorrect names, i.e., issues that significantly affect performance evaluation. To overcome these challenges, we propose a semantic similarity search strategy that can mitigate minor errors and capture conceptually equivalent expressions close to the gold standard. In this context, we introduce new models resulting from the fine-tuning of LLMs integrated with similarity search mechanisms, tailored explicitly for the task of extracting biochemical properties from scientific texts. Based on the conducted experiments in the NatUKE benchmark pipeline, we aim to answer the following research questions (RQ):

RQ 1. Which method is most effective for biochemical property extraction, achieving SoTA results?
RQ 2. Which biochemical properties yield the best and worst overall results, and why?
RQ 3. How competitive are open-source models compared to proprietary paid models?
RQ 4. What is the impact of fine-tuning LLMs on the biochemical property extraction task?
RQ 5. What is the impact of the similarity search strategy on the quality of extraction?
RQ 6. How does the incremental addition of training data affect the performance of LLMs?

We divide the remainder of the article as follows: Section 2 presents related work on automation extraction, graph embeddings, knowledge graphs, named entity recognition, and LLMs. Section 3 presents our proposal for extracting biochemical properties through fine-tuning and similarity search through LLMs. Section 4 presents the experimental evaluation with information about the dataset, experimental setup, results, and discussion. Finally, Section 5 presents the study's conclusions and future work.

2. Related Work

Extracting biochemical information from scientific texts is a challenge in natural language processing (NLP) and bioinformatics. Many approaches have been proposed to automate biochemical knowledge extraction, aiming to transform unstructured data into structured representations to enable large-scale analysis [12]. In this section, we present key advances in areas related to this challenge, with five complementary fronts: automated extraction methods that support the identification of entities and relations; semantic representations based on word embeddings and knowledge graph embeddings, which capture latent meanings and complex relationships between concepts; named entity recognition (NER) techniques, which are essential for the accurate identification of biochemical entities in domain-specific texts; and large language models, state of the art in NLP [11].

2.1. Automation extraction

Recent advancements in language models and information extraction systems have significantly impacted the extraction of knowledge from academic literature, with applications spanning various domains. The Plumber framework [13] represents a significant advancement in this area, providing dynamic information extraction pipelines that are adaptable to diverse data types. This framework is particularly notable for its ability to dynamically create pipelines based on input sentences, as evaluated on DBpedia and ORKG, marking a significant stride in the extraction of structured data. In [14] the authors extended the Plumber framework to 40 reusable components in extraction subtasks like coreference resolution, entity linking, and relation extraction. With this new addition Plumber is capable of dynamically generating 432 distinct pipelines for knowledge extraction, allowing more custom extractions to be automatically tailored.

Complementing Plumber, the T2KG system [15] focuses on transforming unstructured text into structured Knowledge Graphs (KGs). It employs a novel hybrid approach for predicate mapping, enhancing the organization and searchability of knowledge bases. In this evolving landscape, the extension of BERT, KG-BERT [16], plays a pivotal role. BERT's advanced bidirectional representations and fine-tuning capabilities facilitate a deeper understanding of textual data. KG-BERT extends this by treating KG triples as textual sequences for knowledge graph completion, demonstrating a significant improvement in triple classification and link prediction tasks.

The Seq2RDF system [17], with its encoder-decoder framework, effectively bridges the gap between natural language and structured knowledge formats, translating narratives into structured RDF triples. This system enhances utility in various applications by leveraging attention mechanisms and KG embeddings for effective translation. On the other hand, starting from knowledge graphs to extract biochemical information, graph embedding models can also bridge the gap, as they learn representations through graphs.

2.2. Graph embeddings

Word embeddings have witnessed rapid advancements, introducing various methodologies for capturing semantic meaning in both textual and graph-structured data. Word2Vec [18] pioneered the use of neural networks to generate vector representations of words that encapsulate their semantic meaning. They built up their work based on the Skip-gram model, a technique for learning distributed vector representations of words that capture both syntactic and semantic relationships. However, their work is also applicable to the bag-of-words model. Moreover, Word2Vec introduces extensions to enhance vector quality, including subsampling of frequent words for speedup and more regular representations. Models could achieve these substantial improvements on significantly larger datasets with a computationally efficient architecture [18].

DeepWalk [19] extended the concept of WordVec2 to graph structures by employing random walks to generate node embeddings. The approach exhibits scalability as an online learning algorithm that builds incremental results and is easily parallelizable. The concept of DeepWalk involves learning the social representations of a graph's vertices, which is enabled by modeling a stream of short random walks. DeepWalk can be described as a generalization of language modeling. Notably, DeepWalk's representations achieved metrics up to 10% higher than competing methods and, in some cases, outperformed baselines using 60% less training data in 2014.

Node2Vec [20] advanced the technique by employing another random walk strategy, enabling greater adaptability to various graph structures. This algorithmic framework employs a mapping technique that projects nodes into a low-dimensional feature space, aiming to optimize the likelihood of preserving the network neighborhoods of these nodes. They introduce a versatile concept for characterizing a node's network neighborhood and devise a biased random walk procedure that adeptly explores various neighborhoods.

Metapath2vec [21] further specializes in heterogeneous networks by utilizing meta-path-based random walks and skip-gram models to capture nuanced structural and semantic relationships. To employ this approach, they define the heterogeneous neighborhood function of a node, allowing for the skip-gram-based maximization of network probability across various node types. Ultimately, they enhance the optimization process by introducing an effective and efficient heterogeneous negative sampling technique. In the scenario of modeling biochemical properties in knowledge graphs, these methods can be extended to knowledge graphs [6].

2.3. Knowledge Graph Embeddings

The representation of structured knowledge in vector spaces has been considerably advanced through Knowledge Graph Embeddings. TransD proposes a fine-grained approach [22], which represents a named symbol object as two vectors: one for the entity and one for the relation. Generalizing existing models, DistMult [23] provides a framework for learning the representations of entities and relations in knowledge bases and using them to extract logical rules. Providing a linear model based on Tucker decomposition TuckER [24] subsumes several tensor factorization approaches for link prediction. ComplEX [25] performs matrix and tensor factorization with vectors that contain complex values while retaining the mathematical definition of the dot product. Entities in RotatE [26] are represented by complex vectors, while relations are expressed through rotations.

EPHEN [27] maps textual information about events through embeddings and the detailed relationships between these events and their components to a low-dimensional vector space. The method is based on graph regularization to propagate information between events and learn more semantic representations for nodes, allowing the application of similarity-based strategies to perform tasks such as classification, clustering, and link prediction. In this sense, do Carmo et al. [6] extend EPHEN for the extraction of biochemical information, building a knowledge graph (NatUKE Benchmark), and performing link prediction between the article node and a biochemical property node by the learned semantic embeddings to perform the extraction of biochemical information. This strategy outperforms results from other graph embedding methods.

In addition, [8] also proposed a knowledge graph embeddings method based on BFS search in the NatUKE Benchmark. The authors performed a BFS search on the biochemical knowledge graph to generate walks that are input for training a word2vec model. The results significantly improved the extraction of compound and species names, but showed lower performance in the other extraction tasks. In addition to knowledge graph embeddings, named entity recognition methods can be coupled with these methods to enhance information retrieval performance, as the retrieval of information from biochemical compounds is closely related to named entity recognition, i.e., the retrieval of information from names. The random walk method was also addapted for use in RDF Knowledge Graphs in RDF2Vec [28], where the authors implemented a version of Node2vec capable of being used directly with these KGs.

2.4. Named Entity Recognition

Named Entity Recognition (NER) serves as a backbone for information extraction, categorizing named entities in textual data. A comprehensive review of NER development from 1996 to 2006 is given in Nadeau et al.'s survey of named entity recognition and classification [29]. The survey begins with an explanation of the handwritten rules that are being applied in conjunction with semi-supervised and unsupervised machine learning techniques. It predicts a significant advance for the fields of biology and genetics. For biomedical texts, the scispaCy [30] library comes with pre-trained models, offering targeted NER solutions.

Liu et al. [31] propose a hybrid deep learning model based on Bidirectional Encoder Representations from Transformers (BERT), Bidirectional Long Short-term Memory with Conditional Random Field (BILSTM), multi-head attention (MHATT), and conditional random field (CRF) to improve the performance of existing models for

NER. The publication suggests that the use of Transformer over BILSTM should be increased in future projects, as Transformer has a better ability to extract features and fewer training difficulties during this project. Using a naive Bayes classifier approach, Tarasova et al. [32] were able to predict token belongings with an invariant accuracy of 0.96–0.99. For the automatic recognition of entity Marine Natural Products (MNP) published in MNP domain literature, Ma et al [33] suggest a method based on an inflated convolutional neural network and a conditional random field to create a named entity recognition model.

In the NatUKE benchmark, [9] proposed an improvement to the EPHEN method, extracting named entities before representing the texts with the BERT model (initial embedding for article nodes in the EPHEN method). The authors hypothesized that with sentences containing named entities, the model's embedding would be able to extract more information, thereby improving the performance of the EPHEN method, rather than using the first 512 tokens of the article (the default for the BERT method). This strategy showed an improvement in the extraction of bioactivity and isolation-type properties in the Natuke Benchmark. Although performance improved in two biochemical properties, there was no progress in the other properties, leaving a gap in the performance of the methods in the Natuke benchmark. Therefore, this study highlights the need for more robust NLP methods, such as large language models.

2.5. Large Language Models

The introduction of the Transformer model by Vaswani et al. [34] marked a significant advancement in large language models (LLMs). This model fundamentally changed sequence transduction by exclusively relying on attention mechanisms, discarding recurrent and convolutional layers, and leading to streamlined training processes and enhanced performance in tasks such as machine translation. The architecture of the Transformer has since been integral to the development of LLMs, facilitating more efficient and effective NLP techniques.

Building upon this, 'Language Models are Few-Shot Learners' [35] introduced GPT-3, a 175 billion parameter language model. GPT-3 has been shown to effectively perform various NLP tasks with minimal examples and without task-specific fine-tuning, representing a significant advancement in the development of adaptable, general-purpose language systems. "LLaMa: Open and Efficient Foundation Language Models" [36] by Meta AI represents another notable advancement in LLMs. The LLaMa series, which includes models ranging from 7B to 65B parameters, is unique for being trained exclusively on publicly available data. The 13B parameter model in this range surpassed GPT-3 in many benchmarks, despite having 162 billion fewer parameters. Its performance against leading models, such as Chinchilla-70B [37] and PaLM-540B [38], demonstrates the efficacy of public datasets in achieving high-quality results, thereby contributing to broader access to advanced LLMs.

These LLMs have shown potential to improve the extraction of complex biochemical properties from academic texts in the NatUKE benchmark [10, 11]. Their ability to process diverse data structures and perform tasks with minimal supervision aligns with the challenges of distilling specific, high-value information from dense scientific literature. In this sense, [10] explored GPT-3.5 turbo to preprocess and enrich the text used by the knowledge graph embedding methods, thereby improving EPHEN, rather than the traditional method (PyMuPDF) used by EPHEN. The results were improved for the properties' bioactivity, isolation type, and collection site. In the same strategy, the study by [11] also evaluated different text extractors from PDFs and the use of LLMs to represent texts through embeddings. The study concludes that extractors such as Nougat and Grobid generate better results than PyMuPDF. Additionally, another key finding of the study is that BERT embeddings outperform those generated by LLMs for the EHPEN method.

Despite the exploration of LLMs in enriching text extracted from PDFs and representing texts through embeddings, we still highlight the lack of more expressive results for compound names, species names, and collection site properties. Another gap is the use of LLMs to perform the final task of detecting biochemical properties in the text of articles. We emphasize that LLMs can play this role [39]. Another example of using LLMs directly for extraction is found in D'Souza et al. [40], where the authors explore using GPT-4o to directly extract information from species, locations, habitats, and ecosystems to better understand the impact of movement between environments and prevent damage to native species i.e., invasion biology. This study presents interesting results, which also motivated our work. Therefore, in the next section, we present our proposal for biochemical knowledge extraction based on LLMs.

3. Large Language Models for biochemical Knowledge Extraction

Large Language Models (LLMs) have been successfully utilized to enhance performance in various text extraction tasks. Some attempts have tried, with limited usage, to integrate LLMs in the task of biochemical Knowledge Extraction [10, 11]. We believe that LLMs can be used as the center of a biochemical Knowledge Extraction pipeline to improve the state-of-the-art results in NatUKE. Therefore, we created a new method based on fine-tuning and similarity search, as we show in Figure 1. We introduce a part of our data to control the output from the LLM for our task through fine-tuning. In-depth details about our fine-tuning setup are presented in the Subsection 3.1. Since biochemical Knowledge can present many synonyms and different writing standards, the LLMs have difficulty in learning the correct output format. Therefore, we apply a similarity search setup within a common embedding space to transform the LLM output into the closest indexed known answer for the NatUKE benchmark. Details about our similarity search setup are presented in Subsection 3.2. We chose Nougat [41] as a PDF extraction (a tool designed for OCR extraction of academic papers) since Nougat has achieved the state-of-the-art results in the task of biochemical Knowledge Extraction in do Carmo et al. [11].

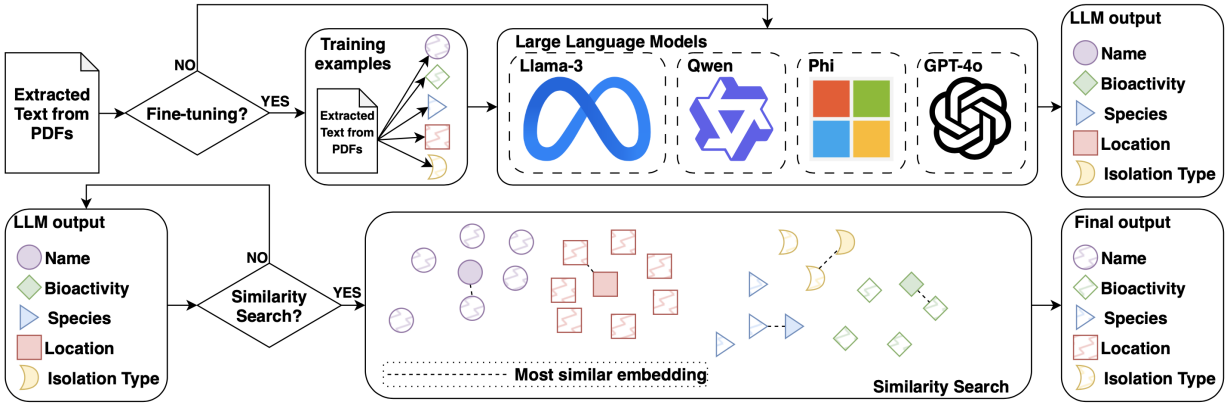


Fig. 1. Our proposal: LLMs for biochemical Knowledge Extraction with fine-tuning and similarity search.

3.1. Finetuning

Fine-tuning is a well-established technique for enhancing performance in specific tasks using pre-trained models. We developed our fine-tuning to better control the output from the LLM and improve the performance. All the data for evaluation comes from the NatUKE benchmark [6] splits. This benchmark was initially designed to evaluate graph embedding-based systems, and all splits were randomly selected and considered different tasks. Therefore, we chose to implement the fine-tuning by training five single-task models [42]. When considering single-task fine-tuning, the primary advantage is that by focusing on a single task, it can better adjust the weights and thereby improve performance more consistently. However, it increases the time and computational effort needed, as the same papers will be analyzed and used for training in different tasks.

In single-task models, every fine-tuning step needs to be executed separately, and it requires specific task prompts. In Figure 2, we present the prompt templates for our open-source and proprietary LLMs. The templates are to be modified for each task and filled in for each paper in the training set. The extracted PDF text is presented in the user prompt. Moreover, the known extractions from the training set are also added to the assistant prompt. For GPT-4o, the variable *[task]* is added accordingly, and a JSON format example is provided with generic answers. For the open-source models, use the *[task explanation]* variable in the prompt, and it refers to a customized sentence added for each task, as follows:

– *compound name*: For each paper, you have to analyze the content (text) to identify the Compound name. It can be more than one compound name.

1.

System prompt:
You are a scientist trained in chemistry. You must extract information from scientific papers identifying relevant properties associated with each natural product discussed in the academic publication. For each paper, *[task explanation]*. Your output should be a list with the names. Return only the list, without any additional information.

User:
[paper text extracted with Nougat]

Assistant:
[Example extracted 1, Example extracted 2]

Fine-tuning prompt template for Llama, Qwen and Phi

2.

System prompt:
You are a chemist expert in natural products. Extract the [task]s from the following text. Provide the answers in JSON format: [{*[task]*: *[Example extracted 1]*}, {*[task]*: *[Example extracted 2]*}]. If the [task]s are not specified, leave it empty like "".

User:
[paper text extracted with Nougat]

Assistant:
[{*[task]*: *[Example extracted 1, Example extracted 2]*}]

Fine-tuning prompt template for GPT-4o

Fig. 2. Fine-tuning prompt templates for system, user, and assistant entries.

- *biological activity*: For each paper, you have to analyze the content (text) to identify the Biological Activity. It can be more than one biological activity.
- *collection species*: For each paper, you have to analyze the content (text) to identify the Collection Species, i.e., the Species from which natural products were extracted. Provide the scientific name, binomial form. The family name can be provided. For example, Tithonia diversifolia, Styrax camporum (Styracaceae), or Colletotrichum gloeosporioides (Phyllachoraceae).
- *collection sites*: For each paper, you have to analyze the content (text) to identify the collection Site, i.e., the place of the collection.
- *collection type*: For each paper, you have to analyze the content (text) to identify the Collection Type, i.e., Collection type of the species.

For our three open-source models, we execute fine-tuning through *unsloth* library[1], an open-source framework for LLM fine-tuning and reinforcement learning. After we transform our prompting template using our training data, we obtain a stream of messages. Each message then contains the completed system, user, and assistant prompts. Unsloth then uses the Quantized LoRA (QLoRA) [43] fine-tuning strategy, which is an even more optimized version of Low-Rank Adaptation (LoRA) [44].

The main idea of the LoRA strategy is to break down the weights matrix into multiple smaller matrices, allowing the fine-tuning backpropagation to be executed on fewer parameters [44]. Another characteristic of LoRA is that it does not directly update the model's weights [44]. QLoRA further optimizes the LoRA process by quantizing the model, allowing memory to flow between CPU and GPU memory to handle memory spikes [43].

[1] https://docs.unsloth.ai

Quantization is the process of reducing an input to a different format, thereby using less memory. For example, QLoRA transforms the 16-bit parameters of the LoRA model into 4-bits [43]. More specifically, QLoRA employs two techniques for achieving 4-bit fine-tuning: NormalFloat (NF4) quantization and Double Quantization [43].

Since QLoRA is a quantized version of LoRA, it is essential to define the objective function of LoRA, which allows it to fine-tune a pre-trained language model $p_\Phi(x|y)$, where Φ represents the parameters. Given a training set of token context-paris $\mathcal{Z} = \{(x_i, y_i)_{i=1,...N}\}$, where both x_i and y_i are sequences of tokens. In our case, x_i would represent paper tokens and y_i would represent the example manually extracted for the desired *task*. LoRA optimizes the fine-tuning process by adopting a parameter-efficient approach using task-specific parameters, kept in much smaller matrices and identified by the function $\Delta\Phi(\Theta)$, thus making the task of optimizing this smaller set of parameters dependent on Θ:

$$\max_{\Theta} \sum_{(x,y)\in\mathcal{Z}} \sum_{t=1}^{|y|} \log\left(p_{\Phi_0+\Delta\Phi(\Theta)}(y_t|x, y_{<t})\right). \tag{1}$$

The task-specific parameter update through the specialized objective function that allows updating the weights of the parameters over a task Θ as to reduce error on the next token prediction task. The task Θ to be optimized is applied to a low-rank decomposition of the fixed weights matrix $W_0 \in \mathbb{R}^{d\times k}$ represented by $W_0 + \Delta W = W_0 + BA$, where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$ when the rank $r << min(d, k)$, meaning:

$$Y = W_0 x + BAx, \tag{2}$$

where BAx represents the weights for the parameters that can be updated by the gradient $\Delta\Phi(\Theta)$ in the next token prediction optimization.

QLoRA then applies the NF4 and a Double Quantization on the weights matrix $W_0 x$ for memory savings, but needs to dequantize the format back to the original 16-bit parameters, as follows:

$$Y^{BF16} = doubleDequant(c_1^{FP32}, c_2^{k-bit}, W_0 x^{NF4}) + BAx^{BF16}, \tag{3}$$

$$doubleDequant(c_1^{FP32}, c_2^{k-bit}, W_0 x^{NF4}) = dequant(dequant(c_1^{FP32}, c_2^{k-bit}), W_0 x^{4-bit} = W_0 x^{BF16}, \tag{4}$$

where ^{BF16}, ^{FP32}, $^{k-bit}$, ^{NF4}, and $^{4-bit}$ represent different quantization formats that can be present in different moments of the QLoRA execution.

For GPT-4o, we utilized their *Supervised fine-tuning* option, which involves providing a training file through the dashboard containing messages for the system, user, and assistant prompts in a JSONL format. The fine-tuning then runs in the background for the selected training hyperparameters. However, OpenAI does not disclose which fine-tuning strategy it uses in its proprietary models.

3.2. Similarity Search

After analyzing the results directly attributed to the LLMs, we noticed that most of the incorrect extractions output similar answers, albeit with synonyms or slightly different formats, on tasks such as collection site extraction. Therefore, inspired by NatUKE's original pipeline, which utilizes graph embedding and similarity of embeddings as link prediction, we decided to implement a new method for similarity search on LLM outputs for NatUKE.

We conducted experiments with three different open-source LLMs and GPT-4o. However, there are some apparent differences in executing Similarity Search with open-source and proprietary LLMs. The Similarity

Search pipeline for GPT leverages its implementation of FAISS [45]. Meanwhile, for open-source models, we constructed a pipeline using Nearest Neighbors for cosine similarity between the previously indexed possible answers and the LLM output.

One clear difference between our proposed similarity search for open-source and proprietary LLMs and previous NatUKE implementations [6] is that we aim not to correct the answer, but rather to eliminate formatting errors from the LLM outputs. In short, all we want to do is find the closest answer to each output by the LLM, and we do not care, for example, if this answer was already chosen, which adds no performance to a $hits@k$ performance metric.

For open-source models, we developed a Nearest-Neighbor cosine similarity search pipeline by indexing possible answers, and subsequently the LLM outputs in the same embedding space. For all these embedding models, our Similarity Search pipeline works in the same way as shown in Algorithm 1. Before execution, we indexed all possible answers as items in I and their embedding vectors V for each embedding model. The LLM predictions are stored in P for each task t, evaluation stage s, fold f, and evaluated model m_n in their `pre-trained` and `fine-tuned` (m_t) forms. We then execute Similarity Search in parallel by dividing P into equal parts of n_jobs, and the output is saved in a file with the same format as the LLM outputs in a directory named by the model type m_t, followed by ss. Each file name is formed by the task, stage, fold, and the embedding model e.

Algorithm 1 Embedding-based Nearest Neighbor Similarity Search

function SIMILARITY_RUN(t, s, f, m_n, m_t, e, I, V, P, n_jobs)
  Initialize shared return dictionary D
  Split P into n_jobs parts
  for each split i in parallel **do**
    Load embedding model e
    for all prediction list p_l in split **do**
      for all prediction p in p_l **do**
        $v_{pred} \leftarrow$ encode(p)
        Retrieve nearest item using `NearestNeighbor`
      end for
    end for
    Store results in $D[i]$
  end for
  Aggregate predictions from D
  Write model output to file $m_t\_ss/t\_s\_f\_e$
end function

For GPT-4o, we combined its pre-trained and fine-tuned version with the Facebook AI Similarity Search (FAISS) [45] library to implement similarity search. FAISS offers several advantages over implementing brute force similarity search pipelines in terms of performance and scalability. However, in our case, the most interesting capability was the seamless integration with the OpenAI API. The OpenAI API support was necessary, as we used the OpenAI embedding model `text-embedding-ada-002`[2] to gauge proprietary performance in the Natural Product extraction task. We sought a consistent performance assessment of proprietary LLMs in biochemical Knowledge Extraction, while also aiming to reduce the monetary costs associated with OpenAI API usage.

Like our open-source similarity search pipeline, FAISS also needs an index of possible answers. On the other hand, FAISS utilizes a Hierarchical Navigable Small World (HNSW) algorithm, which compresses the embedding vectors, thereby making the search stage more efficient. For the retrieval step, it utilizes Approximate Nearest Neighbor (ANN) algorithms to achieve further performance gains.

[2] https://platform.openai.com/docs/models/text-embedding-ada-002

4. Experimental Evaluation

This section presents a comprehensive evaluation of our proposed approach, focusing on three key aspects: dataset and curation, experimental settings, and result analysis. Our research goal is to demonstrate that fine-tuning LLMs with similarity search outperforms existing baseline approaches when applied to the NatUKE benchmark, particularly in the task of extracting biochemical properties from scientific literature. All code, configurations, and scripts used in our experiments are publicly available to ensure transparency and reproducibility of our results[3].

4.1. Dataset and Curation

The NatUKE Benchmark focuses on extracting knowledge about natural products from academic literature using graph embedding methods. It leverages a dataset derived from over two thousand instances of annotated biochemical properties of natural products and evaluates four unsupervised graph embedding techniques: DeepWalk, Node2Vec, Metapath2Vec, and EPHEN. The benchmark emphasizes the utility of graph embeddings for retrieving information already embedded in the knowledge graph structure, while also highlighting the need for context-sensitive data to enhance the quality of extraction. The dataset comprises over two thousand entries, each manually annotated by domain experts in chemistry [6].

These annotations cover five specific biochemical properties from the NuBBEDB database [5] that are essential for training and prediction: compound name (rdfs:label), bioactivity (nubbe:biologicalActivity), species from which the natural products were extracted (nubbe:collectionSpecie), collection site of these species (nubbe:collectionSite), and isolation type (nubbe:collectionType). Table 1 shows the number of distinct values per biochemical property in the dataset and the number of articles with valid annotations. After curation, the dataset comprises 143 articles, featuring 448 compound names, 33 bioactivities, 115 species, 51 collection sites, and five isolation types. Thus, the task consists of predicting one or more values for each of these properties per article. In this setting, the input is a scientific article in PDF format, and the output corresponds to the extracted biochemical properties.

Table 1
Overview of the number of distinct values per biochemical property in the dataset.

Property	Compound Names	Bioactivities	Species	Sites	Isolations	Articles
Number	448	33	115	51	5	143

4.2. Experimental Settings

The first pioneering methods explored in the Natuke benchmarking were graph embedding methods for unsupervised knowledge extraction. A knowledge graph with the paper's DOI as a central node is modeled and connected to extracted characteristics (e.g., name, bioactivity, etc). Thus, each paper, along with its unique compound names, bioactivities, species, sites, and isolations, is a node in the knowledge graph. The authors also extract topics from BERTopic (to connect the graph). To manage BERTopic's token limits, papers are split into sentences. Four primary graph embedding baselines were: EPHEN, Metapath2Vec, DeepWalk, and Node2Vec [6]. These methods were evaluated for their effectiveness in extracting specific biochemical properties of natural products, represented as a knowledge graph with the DOI serving as the central node. We compare our strategy with these four graph embedding methods due to their capability to generate embeddings without requiring a complete knowledge graph, predetermined weights, or ontology. After generating the embeddings for all nodes, the methods explore the task as a link prediction task, considering more than one biochemical property, since we explore the hits@k metric. The strategy was based on the k-nn similarity. Each article is connected to the

[3] https://github.com/chemnet-io/ballena

k embedding of similar biochemical properties. The methods are beneficial for real-world applications where papers are interconnected solely by automatically extracted features [6].

We also compare our methods with the three winning methods from the First International biochemical Knowledge Extraction Challenge (BiKE), which aimed to advance beyond the baseline results established by the NatUKE benchmark [46]. These three methods achieved competitive results using different strategies [8–10]. The method proposed by [8] introduces a new knowledge graph embedding approach that leverages breadth-first search (BFS) over the NatUKE knowledge graph to generate embeddings, following the same link prediction strategy as EPHEN based on embedding similarity to extract biochemical properties. The method by [10] employed ChatGPT 3.5-turbo during the preprocessing stage (PDF text extraction), in conjunction with EPHEN. Finally, the method by [9] incorporated Named Entity Recognition (NER) into the preprocessing pipeline, also in conjunction with EPHEN. Another method compared with our methods was the improved version of EPHEN. In [11], the authors developed EPHEN++, an enhanced variant of EPHEN, by incorporating an alternative PDF text extractor and exploring the use of LLM-based embeddings. After an extensive experimental evaluation, the best-performing configuration combined the Nougat PDF extractor with BERT embeddings. We use these best results in our experimental evaluation.

We selected the open-source models LLaMa 3.1 with 8 billion parameters[4] [47], Phi 4 with 14 billion parameters[5] [48], and Qwen 2.5 with 14 billion parameters[6] [49] to perform fine-tuning. Due to hardware constraints, we limited the input text for fine-tuning to the first 3,000 words of each article. For some folds and biochemical properties, we needed to decrease this number to 2000 or 1500 to allow the execution to proceed. For the name property in fold 5 of the LLaMa model and the species property in fold 6 of the Phi model, we used 2000. For the Qwen model and the type property in fold 0, we executed it with 1500. The rest were executed with 3000. We utilized the unsloth library to fine-tune our open-source models. The model was fine-tuned with a learning rate of $5 \cdot 10^{-4}$ over 25 steps. We selected the GPT-4o (snapshot: `gpt-4o-2024-08-06`)[7], and fine-tuned this model with the following parameters: `Epochs: 3`; `Batch size: 1`; `LR multiplier: 2`; `Seed: 815`. Moreover, we fine-tuned GPT-4o as five different single-task models, since fine-tuning for one task allows for less configuration tuning, which could rapidly increase the costs.

Regarding the embedding model used for the open-source models' Similarity Search, we compared performance with five trending sentence embedding models on Hugging Face[8] and report the best result. More specifically, we collected these models on July 18, 2025. Among the top options, we selected five models that ran without errors when indexing our possible answers on an Nvidia A2 GPU using default configurations. The chosen embedding models were the following[9]: (1) `sentence-transformers/all-miniLM-L6-v2`; (2) `Qwen/Qwen3-Embedding-0.6B`; (3) `BAAI/bge-m3`; (4) `intfloat/multilingual-e5-large`; and (5) `sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2`.

The metric for performance evaluation is hits@k, customized for each feature to account for its unique prediction difficulty, i.e., with different k for each biochemical property. We chose Hits@k due to its versatility in scenarios where k must be adjusted according to task difficulty. In the case of the Natuke benchmark, it was observed that certain properties, such as isolation type, are easier to predict, while others, such as compound name, are significantly more challenging to predict. Therefore, this metric aligns more closely with the evaluation dynamics of the Natuke benchmark. The hist@k can be defined as:

$$\text{Hits@k} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\{r_i \leqslant k\} \tag{5}$$

[4] https://huggingface.co/unsloth/Meta-Llama-3.1-8B-Instruct-bnb-4bit
[5] https://huggingface.co/unsloth/phi-4-unsloth-bnb-4bit
[6] https://huggingface.co/unsloth/Qwen2.5-14B-Instruct-bnb-4bit
[7] https://platform.openai.com/docs/models/gpt-4o
[8] https://huggingface.co/models?library=sentence-transformers
[9] All models can be accessed by typing https://huggingface.co/ followed by the presented name

where N is the total number of articles. r_i is the rank position of the correct answer for the i-th article. $\mathbb{I}\{r_i \leqslant k\}$ is the indicator function that equals 1 if the correct answer is within the top-k positions (i.e., $r_i \leqslant k$), and 0 otherwise. We follow the rule used in NatUKE [6] to choose the final k values for each biochemical property. The authors vary k from 1 to 50, considering values multiples of 5. k was increased until it reached 50 or if a score equal to or greater than 0.50 was achieved.

The evaluation process involves forty stages. First, we vary the train and test sets into four stages. The first split is 20/80%, the second 40/60%, the third 60/40%, and the fourth 80/20%. We have ten holdouts in each stage with different seeds. The dataset adheres to the principles of Findable, Accessible, Interoperable, and Reusable (FAIR) to ensure easy replication and further studies. All experiments and data are publicly accessible under the Apache License 2.0 at https://github.com/AKSW/natuke. The NatUKE benchmark has a structure with five prediction tasks, ten training folds, and four evaluation stages, resulting in a scenario of 200 models. Due to this structure, we trained the LLMs using only the first stage of the benchmark splits. This simplification was necessary to reduce complexity. Thus, the model is always trained with 20% of the data and predicts 80% (1st stage), 60% (2nd stage), 40% (3rd stage), and 20% (4th stage). To mitigate this small gap, we performed an ablation study to demonstrate the impact of training the open-source LLMs across all stages in a single fold.

4.3. Results and Discussion

Considering all our experiments and the NatUKE benchmark, we present Table 2, which contains previous results and our experiments with open-source and proprietary LLMs. The table is divided in blocks related to different types of experiments: (i) the original NatUKE benchmark graph embedding results; (ii) the extended results from the BiKE challenge, and EPHEN increments; (iii) our evaluation of three open-source LLMs with and without our proposed pipeline; and (iv) our GPT-4o evaluation with and without our proposed pipeline. For (iii) and (iv), **SS** means that Similarity Search was applied to a pre-trained LLM output, and **FT-SS** means that Similarity Search was applied to a fine-tuned LLM output, i.e., the entire proposed pipeline was evaluated. Overall, different LLMs obtain the best and second-best performance in all scenarios. Furthermore, we present an in-depth analysis guided by our proposed Research Questions.

4.3.1. Which method is most effective for biochemical property extraction, achieving SoTA results?

Looking at Table 2, we can see that for different tasks, different models and configurations achieve the best and second-best performances. Most notably, we can observe that **GPT-4o** achieves the best and second-best performance with its **SS** and **FT-SS** configurations in the collection species extraction task. However, it would only achieve third-best performance from the LLMs in its pre-trained output. This indicates that most likely **GPT-4o** has seen many species in its pre-trained version and might have made a mistake with species in the same genus or family, and the similarity search was able to find the correct species for the paper. However, fine-tuning reduced performance for species extraction, which can be caused by overfitting some specific tokens, skewing the results down.

GPT-4o also achieved the clear best performance in the collection site, however, with its **FT-SS** configuration this time. Moreover, its performance was so much better than the other LLMs in this task that its **SS** configuration achieved second-best performance. For the collection site, we observe that without our proposed similarity search approach, the pre-trained LLMs fail to understand the formatting required by Nubbe's annotators, as all LLMs achieve virtually 0 $hits@20$. After similarity search, they mostly outperform the original graph embedding-based models from NatUKE.

For compound name, we also achieve a clear new SoTA on **Phi-SS**, followed by **Qwen-SS** achieving second-best results in all but one evaluation stage. The good performance in compound name from open-source LLMs is an interesting behavior. It is our most challenging extraction due to the high number of options compared to $k = 50$, and to the high complexity of scientific names. One clue as to why they achieved such good performance comparatively is when we look at **GPT-4o** achieving $.00$ in all evaluation stages, while all three open-source LLMs achieved $hits@50$ within $.05 \leqslant x \leqslant .10$. This behavior indicates the presence of compound names in the pre-training data of those LLMs, while they might not have been present in **GPT-4o**'s pre-training.

For bioactivity and collection type extraction, our two least challenging extraction tasks, we observe more than one LLM achieving best and second-best results. **Qwen** with both its **SS** and **FT-SS** variants was only around

Table 2

Performance of graph embedding methods (first 4 methods), methods that improved EPHEN's performance (next 4 methods), open-source LLMs (next 12 methods), and proprietary LLM (final 3 methods). We present $hits@k$ for all stages and properties. Best results are in bold and second-best underlined.

	Name (k=50)				Bioactivity (k=5)				Specie (k=50)				Site (k=20)				Type (k=1)			
	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th	1st	2nd	3rd	4th
DeepWalk [6]	.08	.00	.00	.00	.41	.12	.10	.07	.37	.24	.27	.25	.56	.41	.38	.29	.25	.14	.14	.09
Node2Vec [6]	.08	.00	.00	.00	.41	.07	.03	.03	.36	.22	.25	.24	.57	.36	.28	.23	.10	.07	.05	.01
Metapath2Vec [6]	.10	.08	.09	.20	.27	.17	.13	.12	.40	.41	.42	.44	.40	.42	.42	.40	.28	.22	.19	.18
EPHEN [6]	.09	.02	.03	.04	.55	.57	.60	.64	.36	.24	.29	.30	.53	.52	.55	.55	.71	.66	.75	.75
Zope [8]	.09	.19	.35	.83	.37	.11	.11	.10	.47	.65	.75	.81	.32	.31	.36	.41	.03	.04	.05	.12
Fröhlich [10]	.09	.00	.00	.00	.62	.69	.69	.69	.35	.24	.30	.30	.56	.60	.62	.64	.76	.77	.80	.81
Dichte [9]	.09	.00	.00	.00	.60	.65	.65	.69	.34	.23	.27	.29	.55	.58	.60	.57	.73	.74	.77	.76
EPHEN++ [11]	.09	.00	.00	.00	.59	.66	.69	.71	.34	.23	.29	.30	.56	.62	.63	.65	.78	.78	.78	.80
LLaMa	.05	.05	.06	.07	.00	.00	.00	.00	.10	.09	.09	.09	.01	.01	.01	.01	.00	.00	.00	.00
LLaMa-SS	.49	.51	.50	.50	.44	.45	.44	.45	.52	.52	.50	.49	.30	.31	.32	.31	.81	.82	.83	.84
LLaMa-FT-SS	.70	.72	.73	.74	**.74**	.74	**.76**	**.77**	.86	.88	.88	.86	.63	.63	.64	.65	.91	**.92**	**.93**	.94
Qwen	.09	.09	.08	.10	.06	.06	.05	.06	.37	.39	.41	.38	.01	.01	.01	.01	.00	.00	.00	.00
Qwen-SS	.81	.82	.81	.81	.71	.72	.74	.72	.86	.86	.86	.86	.61	.62	.64	.66	.79	.80	.81	.84
Qwen-FT-SS	.62	.64	.65	.67	.71	.73	.75	.76	.89	.91	.92	.92	.62	.63	.64	.66	**.92**	**.92**	**.93**	.94
Phi	.08	.08	.08	.09	.02	.02	.02	.01	.22	.23	.24	.23	.01	.01	.01	.01	.00	.00	.00	.00
Phi-SS	**.85**	**.87**	**.87**	**.89**	.72	.74	.75	.74	.90	.90	.90	.89	.61	.63	.64	.66	.66	.65	.66	.65
Phi-FT-SS	.65	.68	.69	.70	.73	.74	.75	.74	.83	.84	.86	.87	.61	.61	.63	.64	.91	**.92**	**.93**	.94
GPT-4o	.00	.00	.00	.00	.03	.04	.03	.03	.19	.20	.24	.22	.00	.00	.00	.00	.00	.00	.00	.00
GPT-4o-SS	.70	.74	.74	.81	**.74**	**.77**	**.76**	.76	**1.0**	**1.0**	**1.0**	**1.0**	.69	.67	.75	.77	.15	.14	.14	.07
GPT-4o-FT-SS	.60	.59	.60	.63	.68	.68	.67	.73	.94	.96	.94	.92	**.82**	**.77**	**.84**	**.92**	**.92**	**.92**	**.93**	**.96**

.02 $hits@5$ points from the best performers in the bioactivity extraction task. For collection type extraction, all **FT-SS** variants from the LLMs achieved $\approx .95$ $hits@1$ score. This behavior suggests that a ceiling for the LLMs is being reached in these two extraction tasks.

4.3.2. Which biochemical properties yield the best and worst overall results, and why?

When discussing extraction tasks, it is essential first to define the challenges each task poses. In Table 1 we present the number of possible options in the NubbeKG used by the NatUKE benchmark. The higher the number of distinct possible values, the higher the statistical challenge itself. In the original benchmark, the authors proposed a rule to choose the k value for the $hits@k$ metric, depending on which k value surpassed 0.5 or 0.2 score to try and output representative numbers for each one. Moreover, each task has particular challenges regarding specific inputs for different reasons, for example:

- **compound name:** for compound names, they can have complex ways of writing and standards, for example: *IUPAC name* or *common name*;
- **bioactivity:** bioactivity represents the behavior a certain biochemical compound will have when administered in a living being, and there are synonyms that the annotators might have used to standardize the dataset that are different than the one written in the paper;
- **collection species:** scientific names of species are standardized and should be written in the same way by the authors of the paper and the annotators; however, since there are many examples of scientific names, larger models might have an advantage in identifying them due to having large training data;

- **collection site:** for the location where a species was extracted from, annotators have used a particular formatting for the city and state name, which pre-trained models might use differently;
- **collection type:** the annotators standardized the collection types into a minimal set of options, so pre-trained models might not be aware of their standardization.

When looking at Table 2, the first two things we notice are virtually the same collection type extraction performance value for all **FT-SS** LLMs; and (2) the 1.0 results for the species extraction task for **GPT-4o-SS**. The 4-way tie suggests our proposed pipeline most likely reached a ceiling in this task, specially since this is the single task classified with $hits@1$ by the most relaxed NatUKE rule (i.e., whenever a $hits@k$ surpassed 0.5 evaluation score in the k range of 1 to 50 going up by multiples of 5). Moreover, **GPT-4o-FT-SS** achieved the clear best $hits@20$ performance for the collection site extraction task. When considering our open-source LLMs **Phi-SS** obtained the best $hits@50$ performance when extracting the compound name. For bioactivity, we had a performance tie between **LLaMa3-FT-SS** and **GPT-4o-SS**, while **Phi-SS** and **Phi-FT-SS** achieved tied second-best performance, suggesting we are also approaching a ceiling in this extraction task. Another task that LLMs might be reaching a ceiling is on collection species extraction, where **GPT-4o-SS** has the clear best performance on the collection species task, achieving a 1.0 $hits@50$ result in all evaluation stages. We believe two main reasons contribute to this outstanding result, even though the collection species tasks have the second highest number of distinct values: (i) in the original NatUKE benchmark the collection species achieved $hits@50$ by surpassing the 0.5 rule with EPHEN, unlike in the compound name, where the 0.5 scora was not achieved by any model; and (ii) the large amount of training data and parameters for **GPT-4o** means that most likely, all species evaluated in the benchmark have been seen by the model at some point.

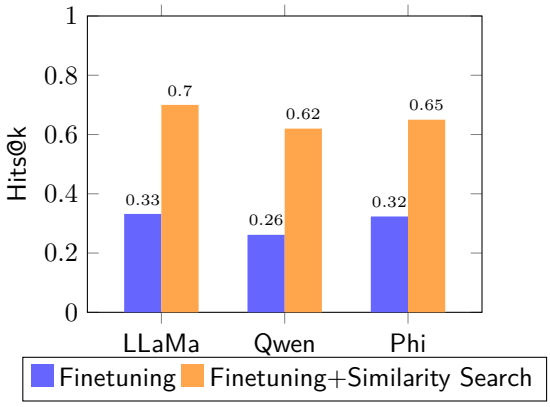
4.3.3. How competitive are open-source models compared to proprietary paid models?

Open-source and proprietary performance in the task of biochemical knowledge extraction depends on which property of task extraction the LLMs are applied to. The open-source LLMs are least competitive in the collection species and site extraction tasks. For the collection species extraction **Qwen-FT-SS** had the best open-source performance by tying **GPT-4o-FT-SS** for second-best in the 4th evaluation stage. However, for collection site extraction, the open-source models had an even bigger performance gap, where best and second-best results were achieved by **GPT-4o**'s **FT-SS** and **SS** variants respectively, and the closest open-source models' performances were $.11$ $hits@20$ behind **GPT-4o-SS** at second-best. Additionally, open-source models showed minimal performance difference between their **SS** and **FT-SS** variants for collection site extraction.

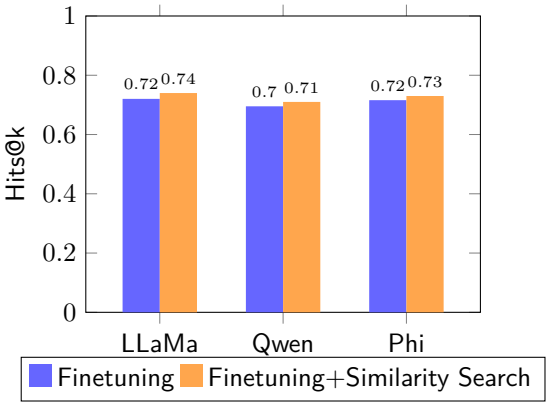
On the other hand, for compound name extraction **Phi-SS** achieved the best performance at $.89$ while **GPT-4o** only managed $.81$ $hits@50$ in the 4th evaluation stage. The gap is even larger at $.15$ $hits@50$ in the 1st evaluation stage, which contains the largest test set. As we discussed, the exceptional performance for compound name extraction can be explained by the content being present in the pre-training stage for the open-source LLMs. Moreover, for the bioactivity and collection type extraction task, the open-source LLMs manage to match the performance of **GPT-4o**, tying results for best and second-best throughout all four evaluation stages.

4.3.4. What is the impact of fine-tuning LLMs on the biochemical property extraction task?

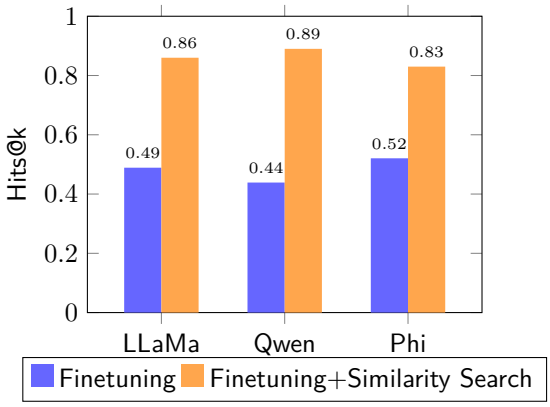
In Figure 3 we present results from our open-source models in their **FT** and **FT-SS** configurations. When looking at the blue bar representing the **FT** variants from LLaMa, Qwen, and Phi, we can see that it allows the models to start correctly recognizing extractions without similarity search. More specifically, for the bioactivity, collection site, and isolation type extraction tasks, **FT** yields similar performance to the **FT-SS** variant. These extraction tasks are particularly noteworthy because they involve a more fixed set of answers standardized by human annotators from Nubbe. Consequently, fine-tuning proved particularly effective in aligning synonyms and different formats with the expected answers before the similarity search step. Meanwhile, for the compound name and collection species extraction tasks, the **FT-SS** yields significantly more performance than just the **FT** variant, and when looking back at Table 2, the **SS** variant achieves the best results with different LLMs. Another characteristic of compound name and collection species extraction is the performance of more than 0.00 $hits@k$ for pre-trained LLMs, indicating that the alignment between annotated and pre-trained knowledge diminishes the effectiveness of fine-tuning. Finally, fine-tuning enables more effective real-world application of automatic



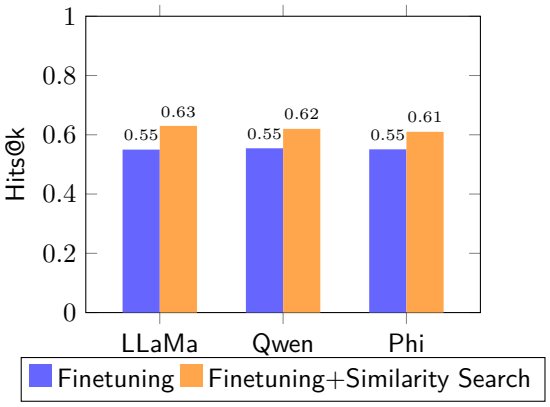
(a) **Compound Name.**



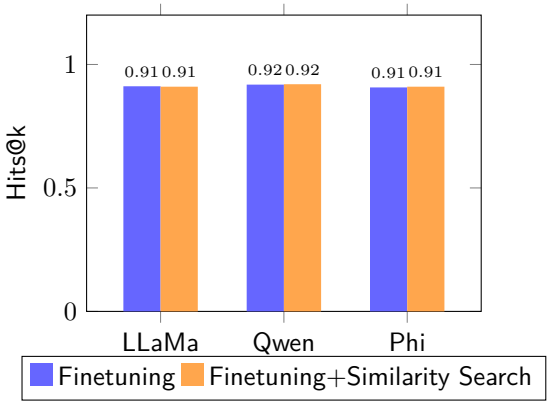
(b) **Biological Activity**.



(c) **Collection Species.**



(d) **Collection Site.**



(e) **Isolation Type.**

Fig. 3. Fine-tuning versus Fine-tuning with similarity search for each model in the 1st stage.

extraction, as it eliminates the need for a predefined indexed list of options, a limitation of previously evaluated graph embedding-based systems that rely on similarity search, while maintaining competitive performance within the NatUKE benchmark.

4.3.5. What is the impact of the similarity search strategy on the quality of extraction?

Our similarity search stage allows correction of the output according to the items present in the database. When looking at Table 2 and Figure 3, we can see that performance increased in all extraction tasks. More specifically, the **SS** variant achieved the best performance in compound name and collection species extraction without fine-tuning. Our experiments demonstrated that these extraction tasks are most susceptible to pre-trained knowledge from LLMs, enabling the correct extraction to be indicated within the $hits@k$ evaluation, particularly when the LLM selects similar compounds or species from the same genus or family. Alternatively, we can observe that for extraction tasks where the annotator's choice of a synonym or format (i.e., bioactivity, collection site, isolation type), the fine-tuning stage can better adapt the LLM output, and the **FT-SS** variants can achieve the best and second-best results more often.

One problem with the usage of similarity search is that it reduces the real-world usage of the automatic extraction pipeline by recreating a limitation of the previous graph embedding-based implementations: it limits the extraction to a known indexed world.

4.3.6. How does the incremental addition of training data affect the performance of LLMs?

For the previous fine-tuning results, only 20% of the data from the first evaluation stage was used due to the increase in processing time and resources required for evaluating fine-tuning on all evaluation stages for every LLM and every fold from NatUKE. In Figure 4, we present the results when using the other evaluation stages' training data for fine-tuning LLaMa. More often than not, there is an increase in performance for training the extra training data present in the fourth evaluation stage in all extraction tasks. In particular, on the collection species extraction task, there is a jump in performance from 0.53 to 0.65 from the second to the third, and another 0.05 $hits@20$ gain by using the fourth evaluation stage training split for fine-tuning. For bioactivity and collection site extraction tasks, the gains are not as significant at 0.05 $hits@5$ and 0.04 $hits@20$ respectively. For compound name and isolation type, our two most different extraction tasks by number of possible answers (see Table 1), there is no notable gain by using more training data for fine-tuning.

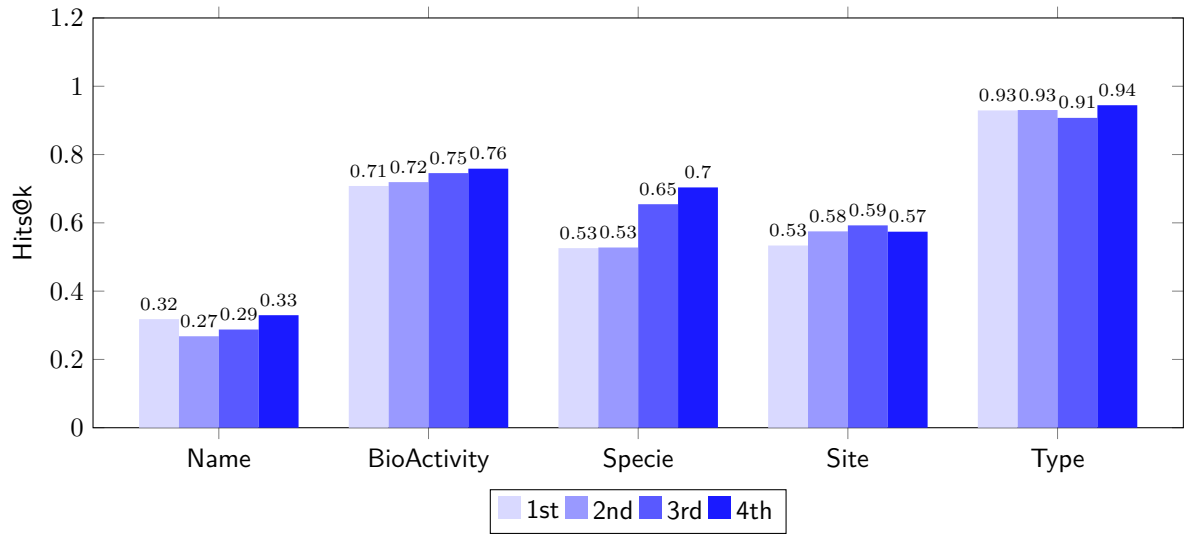


Fig. 4. Fine-tuning of LLaMa in all stages in one fold to evaluate the power of data for biochemical properties prediction.

Additionally, we fine-tuned five single-task models in the entire Nubbe dataset. We made them available on https://huggingface.co/aksw/ under the following model names: `Bike-name`, `Bike-bioactivity`, `Bike-specie`, `Bike-site`, `Bike-isolation`. These models enable the automatic extraction of properties from natural products papers. However, as all the data was used for training, we cannot evaluate them in the NatUKE.

5. Conclusion and Future Work

In this work, we proposed new fine-tuning strategies for large language models (LLMs) combined with similarity search to predict biochemical properties within the NatUKE benchmark. Our research goal was to advance automated biochemical knowledge extraction from scientific literature, contributing to the acceleration of drug discovery based on natural products and addressing limitations observed in previous approaches. We conducted comparative experiments with both open-source and proprietary LLMs, with and without similarity search, evaluating their performance against state-of-the-art methods. Below, we summarize the main answers to our six research questions:

1. **RQ 1 (state-of-the-art results)**. Different strategies performed best for different properties: the pre-trained GPT model combined with similarity search achieved the highest scores for species; the combination of GPT with fine-tuning and similarity search was most effective for the collection site. The Phi model, without fine-tuning but with similarity search, achieved the best results for compound name. In the bioactivity and isolation type tasks, there were ties between different models, such as LLaMa with fine-tuning plus similarity search and pre-trained GPT (bioactivity), and Qwen with fine-tuning plus similarity search and GPT with fine-tuning plus similarity search (isolation type).
2. **RQ 2 (biochemical properties with best and worst overall performance)**. The tasks of predicting species, isolation type, and compound name yielded the best results, with average Hit@k scores above 0.85. In contrast, bioactivity and collection site showed lower performance, with averages ranging from 0.74 to 0.83. These findings underscore the need for future research to enhance performance on the more challenging properties.
3. **RQ 3 (open-source vs. proprietary models)**. Open-source models proved to be competitive and, in some cases, outperformed proprietary ones. They achieved the best results for the compound name and demonstrated comparable performance in terms of bioactivity and isolation type. However, the proprietary GPTs are better in species and collection site predictions.
4. **RQ 4 (impact of fine-tuning)**. Fine-tuning had a positive impact on some models (e.g., Qwen and LLaMa) and a negative impact on others (e.g., GPT and Phi, specifically in certain tasks). Overall, combining fine-tuning with similarity search outperformed using similarity search alone, with notable exceptions—for instance, the Phi model achieved better results with similarity search only.
5. **RQ 5 (impact of similarity search)**. The use of similarity search consistently improved performance, regardless of model or configuration. In all cases, it enhanced Hit@k scores, demonstrating its effectiveness in mitigating subtle textual prediction errors.
6. **RQ 6 (impact of incrementally adding training data to LLMs)**. The incremental addition of training data improved the performance of open-source LLMs in predicting bioactivity, species, and sites. However, no improvement was observed for more difficult tasks, such as predicting the compound name (with 418 unique values), or for the simplest task, isolation type (with only 5 options). These results are directly related to the inherent number of distinct options for each biochemical property.

These results reinforce the potential of using LLMs, whether open-source or proprietary, in combination with similarity search for extracting biochemical properties from scientific articles. Nonetheless, this study has a few limitations. First, our similarity search strategy, although effective, remains naive and may return duplicate values for the same biochemical property. Second, training open-source models requires robust computational infrastructure. Third, applying fine-tuning to proprietary models entails financial costs. Lastly, the current design of the NatUKE benchmark imposes constraints that hinder the broad evaluation under different configurations.

These limitations also point to clear directions for future research. We intend to enhance our similarity search approach by incorporating structural and semantic constraints, and to explore smaller, more accessible open-source models optimized for low-resource environments. We also plan to investigate new fine-tuning strategies, including approaches based on reasoning [50]. Finally, we aim to develop NatUKE 2.0, a new benchmark with an experimental design better suited to the capabilities of modern LLMs, including the systematic execution of baseline methods established in the original NatUKE version.

Acknowledgments

This work was partially supported by CAPES (Process No. 88887.671481/2022-00), CNPq (Process No. 316507/2023-7), and the São Paulo Research Foundation (FAPESP) (Grants 2023/10100-4, 2019/07665-4, and 2013/07375-0). We also acknowledge support from the Google Latin American PhD Fellowship.

References

[1] A.G. Atanasov, S.B. Zotchev, V.M. Dirsch and C.T. Supuran, Natural products in drug discovery: advances and opportunities, *Nature reviews Drug discovery* **20**(3) (2021), 200–216.

[2] L. Zhang, J. Song, L. Kong, T. Yuan, W. Li, W. Zhang, B. Hou, Y. Lu and G. Du, The strategies and techniques of drug discovery from natural products, *Pharmacology & Therapeutics* **216** (2020).

[3] D.J. Newman and G.M. Cragg, Natural products as sources of new drugs over the nearly four decades from 01/1981 to 09/2019, *Journal of natural products* **83**(3) (2020), 770–803.

[4] C. Réda, E. Kaufmann and A. Delahaye-Duriez, Machine learning applications in drug development, *Computational and structural biotechnology journal* **18** (2020), 241–252.

[5] A.C. Pilon, M. Valli, A.C. Dametto, M.E.F. Pinto, R.T. Freire, I. Castro-Gamboa, A.D. Andricopulo and V.S. Bolzani, NuBBEDB: an updated database to uncover chemical and biological information from Brazilian biodiversity, *Scientific Reports* **7**(1) (2017), 7215.

[6] P.V. Do Carmo, E. Marx, R. Marcacini, M. Valli, J.V. Silva e Silva and A. Pilon, NatUKE: A Benchmark for Natural Product Knowledge Extraction from Academic Literature, in: *2023 IEEE 17th International Conference on Semantic Computing (ICSC)*, 2023, pp. 199–203. doi:10.1109/ICSC56153.2023.00039.

[7] C. Stadler, J. Lehmann, K. Höffner and S. Auer, Linkedgeodata: A core for a web of spatial open data, *Semantic Web* **3**(4) (2012), 333–354.

[8] B. Zope, S. Mishra and S. Tiwari, Enhancing Biochemical Extraction with BFS-driven Knowledge Graph Embedding approach., in: *TEXT2KG/BiKE@ ESWC*, CEUR-WS, Greece, 2023, pp. 235–243.

[9] S. Schmidt-Dichte and I.J. Mócsy, Improving Natural Product Automatic Extraction with Named Entity Recognition., in: *TEXT2KG/BiKE@ ESWC*, CEUR-WS, Greece, 2023, pp. 226–234.

[10] P. Fröhlich, J. Gwozdz and M. Jooß, Leveraging ChatGPT API for Enhanced Data Preprocessing in NatUKE., in: *TEXT2KG/BiKE@ ESWC*, CEUR-WS, Greece, 2023, pp. 244–255.

[11] P. Viviurka do Carmo, M.P. Silva Gôlo, J. Gwozdz, E. Marx and R. Marcondes Marcacini, Improving Natural Product Knowledge Extraction from Academic Literature with Enhanced PDF Text Extraction and Large Language Models, in: *Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing*, 2025, pp. 980–987.

[12] Y. Galluzzo, A comprehensive review of the data and knowledge graphs approaches in bioinformatics, *Computer Science and Information Systems* (2024), 27–27.

[13] M.Y. Jaradeh, K. Singh, M. Stocker, A. Both and S. Auer, Better Call the Plumber: Orchestrating Dynamic Information Extraction Pipelines, in: *Web Engineering*, M. Brambilla, R. Chbeir, F. Frasincar and I. Manolescu, eds, Springer International Publishing, Cham, 2021, pp. 240–254. ISBN 978-3-030-74296-6.

[14] M.Y. Jaradeh, K. Singh, M. Stocker, A. Both and S. Auer, Information extraction pipelines for knowledge graphs, *Knowledge and Information Systems* **65**(5) (2023), 1989–2016.

[15] N. Kertkeidkachorn and R. Ichise, T2kg: An end-to-end system for creating knowledge graph from unstructured text, in: *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[16] L. Yao, C. Mao and Y. Luo, KG-BERT: BERT for Knowledge Graph Completion, *arXiv* (2019).

[17] Y. Liu, T. Zhang, Z. Liang, H. Ji and D.L. McGuinness, Seq2rDF: An end-to-end application for deriving triples from natural language text, in: *CEUR Workshop Proceedings*, Vol. 2180, CEUR-WS, 2018.

[18] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado and J. Dean, Distributed representations of words and phrases and their compositionality, *Advances in neural information processing systems* **26** (2013).

[19] B. Perozzi, R. Al-Rfou and S. Skiena, DeepWalk, in: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, 2014. doi:10.1145/2623330.2623732.

[20] A. Grover and J. Leskovec, node2vec: Scalable feature learning for networks, in: *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 2016, pp. 855–864.

[21] Y. Dong, N.V. Chawla and A. Swami, Metapath2vec: Scalable Representation Learning for Heterogeneous Networks, in: *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, Association for Computing Machinery, New York, NY, USA, 2017, pp. 135–144–. ISBN 9781450348874. doi:10.1145/3097983.3098036.

[22] G. Ji, S. He, L. Xu, K. Liu and J. Zhao, Knowledge graph embedding via dynamic mapping matrix, in: *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, 2015, pp. 687–696.

[23] B. Yang, S.W.-t. Yih, X. He, J. Gao and L. Deng, Embedding Entities and Relations for Learning and Inference in Knowledge Bases, in: *Proceedings of the International Conference on Learning Representations (ICLR) 2015*, 2015.
[24] I. Balazevic, C. Allen and T. Hospedales, TuckER: Tensor Factorization for Knowledge Graph Completion, in: *2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing*, Association for Computational Linguistics, 2019, pp. 5184–5193.
[25] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier and G. Bouchard, Complex embeddings for simple link prediction, in: *International conference on machine learning*, PMLR, 2016, pp. 2071–2080.
[26] Z. Sun, Z.-H. Deng, J.-Y. Nie and J. Tang, RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space, in: *International Conference on Learning Representations*, 2019.
[27] P. do Carmo and R. Marcacini, Embedding propagation over heterogeneous event networks for link prediction, in: *2021 IEEE International Conference on Big Data (Big Data)*, IEEE, 2021, pp. 4812–4821.
[28] P. Ristoski and H. Paulheim, Rdf2vec: Rdf graph embeddings for data mining, in: *International semantic web conference*, Springer, 2016, pp. 498–514.
[29] D. Nadeau and S. Sekine, A survey of named entity recognition and classification, *Lingvisticae Investigationes* **30**(1) (2007).
[30] M. Neumann, D. King, I. Beltagy and W. Ammar, ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing, in: *Proceedings of the 18th BioNLP Workshop and Shared Task*, Association for Computational Linguistics, 2019. doi:10.18653/v1/w19-5034. https://aclanthology.org/W19-5034/.
[31] J. Liu, L. Gao, S. Guo, R. Ding, X. Huang, L. Ye, Q. Meng, A. Nazari and D. Thiruvady, A hybrid deep-learning approach for complex biochemical named entity recognition, *Knowledge-Based Systems* **221** (2021), 106958. doi:https://doi.org/10.1016/j.knosys.2021.106958. https://www.sciencedirect.com/science/article/pii/S0950705121002215.
[32] O.A. Tarasova, A.V. Rudik, N.Y. Biziukova, D.A. Filimonov and V.V. Poroikov, Chemical named entity recognition in the texts of scientific publications using the naïve Bayes classifier approach, *Journal of Cheminformatics* (2022).
[33] X. Ma, R. Yu, C. Gao, Z. Wei, Y. Xia, X. Wang and H. Liu, Research on named entity recognition method of marine natural products based on attention mechanism, *Frontiers in Chemistry* **11** (2023).
[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser and I. Polosukhin, Attention is all you need, *Advances in neural information processing systems* **30** (2017).
[35] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell et al., Language models are few-shot learners, *Advances in neural information processing systems* **33** (2020), 1877–1901.
[36] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave and G. Lample, LLaMA: Open and Efficient Foundation Language Models, *arXiv* (2023).
[37] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. de Las Casas, L.A. Hendricks, J. Welbl, A. Clark et al., Training compute-optimal large language models, in: *Proceedings of the 36th International Conference on Neural Information Processing Systems*, 2022, pp. 30016–30030.
[38] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H.W. Chung, C. Sutton, S. Gehrmann et al., Palm: Scaling language modeling with pathways, *Journal of Machine Learning Research* **24**(240) (2023), 1–113.
[39] C. Zhai, Large language models and future of information retrieval: opportunities and challenges, in: *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024, pp. 481–490.
[40] J. D'Souza, Z. Laubach, T.A. Mustafa, S. Zarrieß, R. Frühstückl and P. Illari, Mining for Species, Locations, Habitats, and Ecosystems from Scientific Papers in Invasion Biology: A Large-Scale Exploratory Study with Large Language Models, *arXiv preprint arXiv:2501.18287* (2025).
[41] L. Blecher, G. Cucurull, T. Scialom and R. Stojnic, Nougat: Neural Optical Understanding for Academic Documents, 2023.
[42] M. Gozzi and F. Di Maio, Comparative Analysis of Prompt Strategies for Large Language Models: Single-Task vs. Multitask Prompts, *Electronics* **13**(23) (2024), 4712.
[43] T. Dettmers, A. Pagnoni, A. Holtzman and L. Zettlemoyer, QLoRA: Efficient Finetuning of Quantized LLMs, in: *Advances in Neural Information Processing Systems*, Vol. 36, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt and S. Levine, eds, Curran Associates, Inc., 2023, pp. 10088–10115.
[44] E.J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, W. Chen et al., Lora: Low-rank adaptation of large language models., *ICLR* **1**(2) (2022), 3.
[45] M. Douze, A. Guzhva, C. Deng, J. Johnson, G. Szilvasy, P.-E. Mazaré, M. Lomeli, L. Hosseini and H. Jégou, The faiss library, *arXiv preprint arXiv:2401.08281* (2024).
[46] E. Marx, BiKE: First International Biochemical Knowledge Extraction Challenge, 2023. ihttps://aksw.github.io/bike/.
[47] Meta, The Llama 3 Herd of Models, 2024. https://arxiv.org/abs/2407.21783.
[48] M. Abdin, J. Aneja, H. Behl, S. Bubeck, R. Eldan, S. Gunasekar, M. Harrison, R.J. Hewett, M. Javaheripi, P. Kauffmann et al., Phi-4 technical report, 2024.
[49] T. Alibaba, Qwen2 technical report, 2024.
[50] J. Huang and K.C.-C. Chang, Towards Reasoning in Large Language Models: A Survey, in: *61st Annual Meeting of the Association for Computational Linguistics, ACL 2023*, Association for Computational Linguistics (ACL), 2023.